Opinion Mining with Sentiment Graph

Qi Zhang, Yuanbin Wu, Yan Wu, Xuanjing Huang
School of Computer Science
Fudan University
Shanghai, P.R.China
$\{$*qz, ybwu, 10210240075, xjhuang*$\}$*@fudan.edu.cn*

***Abstract*—Opinion mining became an active research topic in recent years due to its wide range of applications. A number of companies offer opinion mining services. One problem that has not been well studied so far is the representation model. In this paper, we propose a novel sentence level sentiment representation model. By taking the observation that lots of sentences which have complicated opinion relations can not be represented well by slots filling or feature-based model, the novel representation model *sentiment graph* is described in this paper. A supervised structural learning method is presented and used to construct sentiment graphs from sentences. Experimental results in a manually labeled corpus are given to show the effectiveness of the proposed approach.**

***Keywords*-Opinion Mining; Sentiment Graph; Structural learning method;**

I. Introduction

Sentiment analysis, which is also called opinion mining, have been received attentions extensively in recently years. Existing fine-grained analyzing approaches usually assumed that opinion units were composed by a number of elements (e.g. product feature, opinion expression, opinion holder, etc.) and determined whether two elements had certain predefined relationship. A typical relation is the connection between a product feature and the opinion expression which comment it [3], [5], [6].

However, through data analysis, we find that lots of sentences do not follow the assumption used by them and evaluations of them can not be correctly expressed based on those representation methods. Consider the following examples, which are extracted from the real online customer reviews:

Example 1: *The interior is a bit noisy on the freeway*[1].

Example 2: *Takes good pictures during the daytime. Very poor picture quality at night*[2].

Based on the definitions of opinion unit proposed by Hu and Liu [3] or Wu et al.[6], from the first example what we can get is only that the reviewer gave his opinion about "interior" and used the opinion word "noisy", which is a negative one. However, the important condition "on the freeway" can not be expressed with those definitions and representations. Without that, the user's opinion is wrongly enlarged to all conditions. The second example is similar with the first one. If the conditions "during the daytime" and "at night" are ignored, the extracted elements and relations could not represent the user's actual opinions. The examples show the simple elements and relation definitions fail to precisely represent opinions.

To address these problems, this paper describes a novel sentiment graph representation. The vertexes in the representation include target, opinion expression, opinion holder,etc. Edges between vertices represent relations between them such as: coordination, contrast, reason, etc. Head and tail are defined for each relation. The whole structure of graph not only contains the individual relations, but also a deeper semantic connections among those relations. With the graph, we propose a supervised structure learning method to convert sentences to this representation

II. Sentiment Graph

This section describes the proposed sentiment representation approach *Sentiment Graph*.

For an opinion bearing sentence, an expressing of opinion is exhibited through a number of text spans which can be very complicate themselves, but very simple in their effect on the opinion. We define those text spans as opinion elements which are the vertexes in sentiment graph. Vertexes are various in text length, syntactic structure etc.. For example, a vertex can be a single noun word as an opinion target, can be an idiom as an opinion expression, even a clause as the constrain condition of certain opinion. Besides individual elements, the semantic connections among them are also important for us to represent an opinion. The labeled edges in sentiment graph are used to catch those relations. Table I shows the definitions and examples of elements. The relations are defined in the Table II.

III. System Description

To represent a sentence with sentiment graph, we need to extract a number of text spans which are candidate vertexes and build the relations among them to get a graph structure. In this work, we focus on the second task, and assume that the graph vertexes have been correctly collected in following formulations of the problem. However, the first task can be

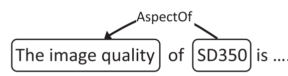
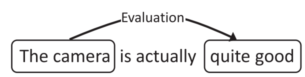
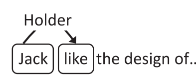
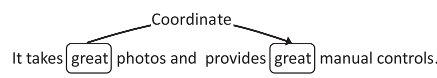
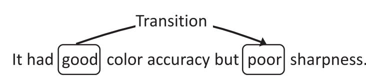
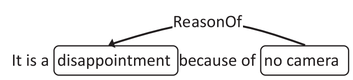
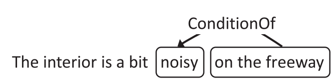
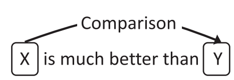
[1]http://reviews.carreview.com/blog/2010-ford-focus-review-the-compact-car-that-can/

[2]http://www.dooyoo.co.uk/digital-camera/sony-cyber-shot-dsc-s500/1151680/

Table I
DEFINITIONS OF ELEMENTS/VERTICES IN SENTIMENT GRAPH.

Element	Definition	Examples
Opinion Target	The subject which the evaluation is focused on.	Named entity (*digital camera model name*, *company name*, etc.) Event (*proliferation of H1N1 flu*).
Aspect	An attribute, a part, or specific features of opinion target	*size*, *color*, *engine*, *noise*, etc.
Opinion Expression	An subjective phrase or a clause used to express an evaluation	*perfect*, *good*, *like*, *recommend*, etc.
Opinion Holder	The person who makes the evaluation	Besides the article's *author*, opinion holders can also be *people*, *organizations*, or even *countries*.
Reason	A span of text which expresses the reason behind an evaluation	Either indicated by a reason conjunctions or by an opinion expression. E.g: *the new lens* improves picture qualities.
Condition	A phrase or a clause which describes situations, status or conditions of the evaluation	*for outdoors*, *on the freeway*, *at night*, etc.

Table II
DEFINITIONS OF RELATIONS/EDGES IN SENTIMENT GRAPH.

Relation	Definition	Example
AspectOf	The relationship between opinion target and aspect. $Target \rightarrow Aspect$	
Evaluation	The relationship between opinion target and opinion expression. $Aspect \rightarrow Expression$	
Holder	The relationship between evaluation and opinion holder. $Holder \rightarrow Expression$	
Coordinate	Coordinate relationship between evaluations. $Expression \rightarrow Expression$	
Transition	Transition relationship between evaluations. $Expression \rightarrow Expression$	
ReasonOf	Reason of the evaluation. $Reason \rightarrow Expression$	
ConditionOf	Condition of the evaluation. $Condition \rightarrow Expression$	
Comparison	Comparative relationship between opinion targets. $Target \rightarrow Target$	

considered as a sequential labeling task as in named entity recognition.

A. *Definitions*

In this section, we begin to formulate the process of constructing a sentiment graph into the framework of structure learning which has been successfully used in other NLP tasks. From now on, a sentence is denoted by s, $\mathbf{x}$ are text spans which will be graph vertexes, x_i is ith vertex in $\mathbf{x}$ subscripted by their positions in s. For a set of vertexes $\mathbf{x}$, $\mathbf{y}$ is the sentiment graph correspond to it and $e = (x_i, x_j)$ is the direct edge from x_i to x_j in $\mathbf{y}$, $\mathcal{L}$ is the edge label set.$\mathcal{G} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_n^N$ denotes vertexes and graph pairs for training.

Follow the edge based factorization in [2],the score of a sentiment graph $s(\mathbf{x}, \mathbf{y})$is the sum of the scores in its edges.

$$s(\mathbf{x}, \mathbf{y}) = \sum_{(x_i,x_j)\in\mathbf{y}} s(x_i, x_j) = \sum_{(x_i,x_j)\in\mathbf{y}} \alpha \cdot \mathbf{f}(x_i, x_j).$$

for each edge(x_i, x_j) and $l \in \mathcal{L}$, the score is a linear function of a high dimensional feature vector $\mathbf{f}(x_i, x_j, l)$ which takes binary value.For example,

$$\mathbf{f}(x_i, x_j, l) = \begin{cases} 1 \ \texttt{if} \ x_i.POS = \texttt{JJ} \ \texttt{and} \ \ x_j.POS = \texttt{NN} \\ \qquad \texttt{and} \ l = \texttt{AspectOf} \\ 0 \ \ \textit{otherwise} \end{cases}$$

When omits the last argument, $\mathbf{f}(x_i, x_j) = \max_{l \in \mathcal{L}} \mathbf{f}(x_i, x_j, l)$.

B. Inference

The learning process aims to get the parameter α which will assign the correct sentiment graph $\mathbf{y}$ with the highest score among all possible graphs of $\mathbf{x}$(denoted by $\mathcal{Y}$).

$$\mathbf{y} = \arg\max_{\mathbf{y} \in \mathcal{Y}} s(\mathbf{x}, \mathbf{y}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \sum_{(x_i, x_j) \in \mathbf{y}} \alpha \cdot \mathbf{f}(x_i, x_j). \quad (1)$$

Like other structure learning tasks, the "$\arg\max$" operation in the equation, also called inference,is hard because all possible values of $\mathbf{y}$ form a huge search space. To find the $\mathbf{y}$ with the largest score, we use structure constrain of the output to reduce the computational complexity.

C. Training

We use the averaged percetron for training [1]. The pseudo code is below:

Averaged Perceptron
Training Set:$\mathcal{G} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_n^N$

$\alpha^0 = 0, i = 0, r = 0$
for $t = 0$ to T **do**
  for $n = 0$ to N **do**
    $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} s(\mathbf{x}_n, \mathbf{y})$
    if $\hat{\mathbf{y}} \neq \mathbf{y}_n$ **then**
      $\alpha^{i+1} = \alpha^i + \sum_{(x_i, x_j) \in \mathbf{y}_n} \mathbf{f}(x_i, x_j) - \sum_{(x_i, x_j, x) \in \hat{\mathbf{y}}} \mathbf{f}(x_i, x_j)$
      $r = r + \alpha^{i+1}$
      $i = i + 1$
    end if
  end for
end for
return $r/(N * T)$

Inside Features	Outside Features
x_i.text	word bigram before
word/POS bigram prefix	POS bigram before
word/POS bigram suffix	word bigram after
dose x_i have digital	POS bigram after
does x_i have sentiment word	char bigram before
is x_i a single word	char bigram after
dose x_i have a parallel phrase	

Table III
FEATURE SET

D. Feature Construction

For each vertex x_i in a sentiment graph, we use 2 sets of features: inside features which are extracted in the text span of x_i, and outside features which are around the text span of x_i. Due to the lack of labeled sentences, we also use an external sentiment lexicon which contains hundreds of sentiment bearing words. And in order to involve syntactic information, the dependency tree of a sentence also used as a feature. The features are list in Table III. Finally, the high dimensional feature vector $\mathbf{f}(x_i, x_j)$ which appears in the learning algorithm is generated by running the binary feature functions over the feature sets of x_i and x_j.

IV. EXPERIMENTS

A. Corpus

We constructed a Chinese online review corpus from Pcpop.com, Zol.com.cn, and It168.com. The corpus makes up by 138 documents, which totally contain 1735 sentences. Since a number of sentences do not contain any opinions, 1364 subjective sentences are finally chosen and each sentence was manually labeled with a sentiment graph. Two annotators labeled the corpus independently. The annotators started from identifying elements. Then for each opinion expression, they annotated the other elements which have relation with it.

B. Experiments Configurations

For supervised learning, we take 90% of the corpus as training set, 10% as test set. In feature construction, we use an external Chinese sentiment lexicon which containing 4566 positive opinion words and 4370 negative opinion words, FDU-NLP tools for Chinese word segment, and Stanford parser [4] as dependency parser. In the settings of averaged perceptron training algorithm, the maximum iteration number is set to 10 which is chosen by maximizing the testing performances.

C. Results

1. The importance of structure information. An alternative method to extract relations is directly using a classifier to judge whether there is a relation between two elements. Those kinds of methods are used in previous opinion mining works [6], [5]. To show the entire structure information is important for the mining relations, we construct an SVM for binary classification of elements pairs. The data point representing an element pair (x_i, x_j) is the same as the high dimensional feature vectors $\mathbf{f}(x_i, x_j)$. The results are shown in the Table 5:

One reason for the poor performance of binary classifier is the huge unbalance on positive and negative training samples(only $\Theta(n)$ positive pairs among all n^2 pairs). And the absent of global structure information makes binary classifier can not catch the relations among results of different elements pairs.

	P	R	F
SVM	64.9	24.0	35.0
Structure	41.5	61.4	49.5

Table IV
BINARY CLASSIFIER AND STRUCTURE LEARNING

2. The effect of inference algorithm. In the inference algorithm, we utilize the properties of sentiment graph and divide the inference process into 3 steps. The opinion bone (Property 1) is an additional constrain compared with direct maximum spanning tree inference. To evaluate whether the system suffers from this additional constrain, we implement a system directly using maximum spanning tree as the inference results, and compare it with our system by omitting the last inference step which also results a tree.

	P	R	F
MST	52.2	50.0	51.1
OBMST	52.2	49.9	51.0

Table V
RESULTS COMPARING INFERENCE METHODS

As Table 6 shows, the direct maximum spanning tree inference performs nearly the same as our system, but takes $O(n^2)$ in time complexity. Thus the property of opinion bone is helpful in getting an enough accurate sentiment graph. However, as mentioned in the previous section, the dynamic programming method for finding opinion bone tends to line all opinion expressions, we think that is the main reason why the method doesn't achieve better results.

3. The effects of various features. We evaluate the performances of different feature configurations. In Table 7, "In" represents the result of inside feature set; "In-s" is "In" without the external opinion lexicon feature; "Out" uses the outside feature set; "In+Out" uses both "In" and "Out", "In+Out+Dep" adds the dependency feature.

	P	R	F
In-s	36.9	49.4	42.2
In	38.9	50.5	43.9
Out	39.8	59.9	47.8
In+Out	41.7	60.1	49.2
In+Out+Dep	41.5	61.4	49.5

Table VI
RESULTS COMPARING DIFFERENT FEATURES

From the results, we observe that the outside feature set is more effective than inside feature set, even if it doesn't use any external resource. A possible reason for this is that the content of a vertex can be very complicated (a vertex even can be a clause), but the context surrounding the vertex is relatively simple and easy to identify. The dependency feature has limited effect, due to the most of online review sentences doesn't obey grammar, the parsing results are unreliable. And also the complexity of vertexes messes the dependency feature.

V. CONCLUSIONS

This paper introduces a novel sentiment representation approach using sentiment graph and proposes a structure learning model to convert sentences to this representation. The main advantages of the proposed sentiment graph are that: 1) opinion elements and relations are complete and can be easily expanded; 2) nested structures are allowed. Based on the properties of the sentiment graph, we propose an efficient inference method, which achieves $O(n^2)$ in time complexity. Experimental evaluations with a manually labeled corpus are given to show the importance of structure information, the effectiveness of the proposed approach, and the effects of various features.

VI. ACKNOWLEDGEMENTS

The author wishes to thank the anonymous reviewers for their helpful comments. This work was partially funded by 973 Program (2010CB327906), National Natural Science Foundation of China (61003092, 61073069), 863 Program of China (2009AA01A346), Shanghai Science and Technology Development Funds(10dz1500104), Doctoral Fund of Ministry of Education of China (200802460066), Shanghai Leading Academic Discipline Project (B114), and Key Projects in the National Science & Technology Pillar Program(2009BAH40B04).

REFERENCES

[1] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. pages 1–8, 2002.

[2] J. Eisner. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pages 340–345, Copenhagen, August 1996.

[3] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM.

[4] D. Klein and C. D. Manning. Fast exact inference with a factored model for natural language parsing. In *In Advances in Neural Information Processing Systems 15 (NIPS*, pages 3–10. MIT Press, 2003.

[5] N. Kobayashi, K. Inui, and Y. Matsumoto. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of EMNLP-CoNLL 2007*, 2007.

[6] Y. Wu, Q. Zhang, X. Huang, and L. Wu. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1541, Singapore, August 2009. Association for Computational Linguistics.